I. Lyapunov Exponent of Quadratic Function -- Source Code

```
>restart;
```

Calculate the Lyapunov exponent for f(x) = r*x*(1-x)

Use r = 3.80, 3.81, 3.82, 3.83, 3.84, 3.85, 3.86, 3.87, 3.88, 3.89.
Use the initial condition for intial APACHE III probability of mortality (example 50%) x0 := 0.5.
The variable

s = s(x0,i) := ln(|df(xi)|) + s(i-1) as the sum of the logarithms of the first i iterates.

We take s(x0,0) = 0, so when we take x0 = 0.5 we do not calculate the derivative at 0.5.
The variable

h = h(x0,i) := s(x0,i)/(i+1)

is the estimate of the Lyapunov exponent starting with the initial condition x0 after i iterates.
(In the text, we write the dependence of s and h on the initial condition x0 and the iterate i,
but in the calculation we only use the variables s and h.)

For the first n1=20000 iterates the calculation for h(x0,i) is not printed out.
For the next n2 = 20 iterates the calculation for h(x0,i) is printed out, which is the estimate for the
Lyapunov exponent.

f The function f(x) = r*x*(1-x)
df The derivative df(x) = f'(x)
r: Parameter value.
n1 The number of iterates for which the estimate of h(x0) is not displayed
n2 The number of iterates for which the estimate of h(x0) is displayed
x0: The initial condition

```
>r := 3.81;
f:= x -> r*x*(1-x);
df := x -> r - 2*r*x;
n1  := 20000;
n2 := 100;
>x0 := 0.5:
x := x0:
s := 0:
for i from 1 to n1 do
   x := f(x);
   s := s + ln(abs(df(x))):
od:
printf(`\n\t r = %1.4f \n\n`, r);
printf(`\t x0 = %1.4f \n\n`, x0);
printf(`\t i \t\t h(x0,i) \n`);
for j from 1 to n2 do
   x := f(x);
   s := s + ln(abs(df(x))):
   h := s/(j+n1):
   printf(`\t %d \t %1.4f \n`, j+n1, h );
od:
>r := 3.81;
f:= x -> r*x*(1-x);
df := x -> r - 2*r*x;
n1  := 20000;
n2 := 100;
>x0 := 0.2:
x := x0:
s := 0:
for i from 1 to n1 do
   x := f(x);
   s := s + ln(abs(df(x))):
od:
printf(`\n\t x0 = %1.4f \n\n`, x0);
printf(`\t i \t\t h(x0,i) \n`);
for j from 1 to n2 do
   x := f(x);
   s := s + ln(abs(df(x))):
   h := s/(j+n1):
   printf(`\t %d \t %1.4f \n`, j+n1, h );
od:
>r := 3.81;
f:= x -> r*x*(1-x);
df := x -> r - 2*r*x;
n1  := 20000;
n2 := 100;
>x0 := 0.3:
x := x0:
s := 0:
for i from 1 to n1 do
   x := f(x);
   s := s + ln(abs(df(x))):
od:
printf(`\n\t x0 = %1.4f \n\n`, x0);
printf(`\t i \t\t h(x0,i) \n`);
for j from 1 to n2 do
   x := f(x);
   s := s + ln(abs(df(x))):
   h := s/(j+n1):
   printf(`\t %d \t %1.4f \n`, j+n1, h );
od:
```

```
>r := 3.81;
f:= x -> r*x*(1-x);
df := x -> r - 2*r*x;
n1  := 20000;
n2 := 100;
>x0 := 0.49:
x := x0:
s := 0:
for i from 1 to n1 do
   x := f(x);
   s := s + ln(abs(df(x))):
od:
printf(`\n\t x0 = %1.4f \n\n`, x0);
printf(`\t i \t\t h(x0,i) \n`);
for j from 1 to n2 do
   x := f(x);
   s := s + ln(abs(df(x))):
   h := s/(j+n1):
   printf(`\t %d \t %1.4f \n`, j+n1, h );
od:
>
```